30 Using ontologies to interlink linguistic annotations and improve their accuracy

Antonio Pareja-Lora[1]

Abstract

For the new approaches to language e-learning (e.g. language blended learning, language autonomous learning or mobile-assisted language learning) to succeed, some automatic functions for error correction (for instance, in exercises) will have to be included in the long run in the corresponding environments and/or applications. A possible way to achieve this is to use some Natural Language Processing (NLP) functions within language e-learning applications. These functions should be based on some truly reliable and wide-coverage linguistic annotation tools (e.g. a Part-Of-Speech (POS) tagger, a syntactic parser and/or a semantic tagger). However, linguistic annotation tools usually introduce a not insignificant rate of errors and ambiguities when tagging, which prevents them from being used 'as is' for this purpose. In this paper, we present an annotation architecture and methodology that has helped reduce the rate of errors in POS tagging, by making several POS taggers interoperate and supplement each other. We also introduce briefly the set of ontologies that have helped all these tools intercommunicate and collaborate in order to produce a more accurate joint POS tagging, and how these ontologies were used towards this end. The resulting POS tagging error rate is around 6%, which should allow this function to be included in language e-learning applications for the purpose aforementioned.

Keywords: ontology, interoperability, POS tagging, accuracy, linguistic annotation, tools.

1. Universidad Complutense de Madrid / ATLAS (UNED), Madrid, Spain; apareja@sip.ucm.es

How to cite this chapter: Pareja-Lora, A. (2016). Using ontologies to interlink linguistic annotations and improve their accuracy. In A. Pareja-Lora, C. Calle-Martínez, & P. Rodríguez-Arancón (Eds), *New perspectives on teaching and working with languages in the digital era* (pp. 351-362). Dublin: Research-publishing.net. http://dx.doi.org/10.14705/rpnet.2016.tislid2014.447

1. Introduction

Some of the most recent and interesting approaches to language e-learning incorporate an NLP module to provide the learner with, for example, "exercises, self-assessment tools and an interactive dictionary of key vocabulary and concepts" (Urbano-Mendaña, Corpas-Pastor, & Mitkov, 2013, p. 29). For these approaches to succeed, the corresponding NLP module must be based on some truly reliable and wide-coverage linguistic annotation tools (e.g. a POS tagger, a syntactic parser and/or a semantic tagger). However, "linguistic annotation tools have still some limitations, which can be summarised as follows:

> (1) Normally, they perform annotations only at a certain linguistic level (that is, morphology, syntax, semantics, etc.).
>
> (2) They usually introduce a certain rate of errors and ambiguities when tagging. This error rate ranges from 10% up to 50% of the units annotated for unrestricted, general texts" (Pareja-Lora, 2012b, p. 19).

The interoperation and the integration of several linguistic tools into an appropriate software architecture that provides a multilevel but integrated annotation should most likely solve the limitations stated in (1). Besides, integrating several linguistic annotation tools and making them interoperate can also minimise the limitation stated in (2), as shown in Pareja-Lora and Aguado de Cea (2010).

In this paper, we present an annotation architecture and methodology that (1) unifies "the annotation schemas of different linguistic annotation tools or, more generally speaking, that makes [a set of linguistic] tools (as well as their annotations) interoperate; and (2) [helps] correct or, at least, reduce the errors and the inaccuracies of [these] tools" (Pareja-Lora, 2012b, p. 20). We present also the ontologies (Borst, 1997; Gruber, 1993) developed to solve this interoperability problem. As with many other interoperability problems, they have really helped integrate the different tools and improve the overall performance of the resulting NLP module. In particular, we will show how

we used these ontologies to interlink several POS taggers together, in order to produce a combined POS tagging that outperformed all the tools interlinked. The error rate of the combined POS tagging was around 6%, whereas the error rate of the tools interlinked was around 10%–15%.

2. The annotation architecture

The annotation architecture presented here belongs in the OntoTag's annotation model. This model aimed at specifying:

> "a hybrid (that is, linguistically-motivated and ontology-based) type of annotation suitable for the Semantic Web. [Hence, OntoTag's tags had to] (1) represent linguistic concepts (or linguistic categories, as they are termed within [ISO TC 37]), in order for this model to be linguistically-motivated[2]; (2) be ontological terms (i.e. use an ontological vocabulary), in order for the model to be ontology-based; and (3) be structured (linked) as a collection of ontology-based <Subject, Predicate, Object> triples, as in the usual Semantic Web languages (namely RDF(S) and OWL), in order for the model to be considered suitable for the Semantic Web" (Pareja-Lora, 2012b, p. 20).

Besides, as discussed above, it should be able to merge the annotation of several tools, in order to POS tag texts more accurately (in terms of precision and recall) than some tools available (e.g. Connexor's FDG, Bitext's DataLexica).

Thus, OntoTag's annotation architecture is, in fact, the methodology we propose to merge several linguistic annotations towards the ends mentioned above. This annotation architecture consists of several phases of processing, which are used to annotate each input document incrementally. Its final aim is to offer automatic, standardised, high quality annotations.

2. see http://www.iso.org/iso/standards_development/technical_committees/other_bodies/iso_technical_committee.htm?commid=48104, and also http://www.isocat.org

Briefly, the five different phases of the annotation architecture are (1) distillation, (2) tagging, (3) standardisation, (4) decanting, and (5) merging. Yet, this last phase is sub-divided into two intertwined sub-phases: combination, or intra-level merging, and integration, or inter-level merging. They are described below, each one in a dedicated subsection.

2.1. Distillation

Most linguistic annotation tools do not recognise formatted (marked-up) text as input for annotation; hence, most frequently, the textual information conveyed by the input files (e.g. HTML, Word or PDF files) has to be distilled (extracted) before using it as input for an already existing linguistic annotation tool. The input of this phase is, thus, an unformatted document, consisting of only the textual information (the distilled, plain or clean text) of the input file to be annotated.

2.2. Tagging

In this phase, the clean text document produced in the distillation phase is inputted to the different annotation tools assembled into the architecture. It does not matter at this point the levels or the formats of the output annotations; it is left to the remaining phases of the architecture to cope with these issues. After this phase, the clean text document will be tagged or annotated (1) at a certain (set of) level(s), and (2) according to a tool-dependent annotation scheme and tagset.

2.3. Standardisation

In order for the annotations coming from the different linguistic annotation tools to be conveniently compared and combined, they must be first mapped onto a standard or guideline-compliant – that is, standardised – type of annotation, so that (1) the annotations pertaining to the same tool but to different levels of description are clearly structured and differentiated (or decanted, in OntoTag's terminology), (2) all the annotations pertaining to the same level of description but to different tools use a common vocabulary to refer to each particular

phenomenon described by that level, and (3) the annotations pertaining to different tools and different levels of description can be easily merged later on in a one and unique overall standardised annotation for the document being processed.

It is at this point where OntoTag's ontologies play a crucial role. They have been developed following the existing standards, guidelines and recommendations for annotation (see some details about them below). Accordingly, annotating with reference to OntoTag's ontologies produces a result that uses a standardised type of tagset. For this reason, the tagsets and the annotations from each and every tool are mapped onto the terms of OntoTag's ontologies. Then, after this phase has been applied, all the tags are expressed according to a shared and standardised vocabulary. In addition, this vocabulary can also be considered formal and fully semantic from a computational point of view, since it is referred to ontologies. The level-driven, taxonomical and relational structure of OntoTag's ontologies is also right and proper for (1) structuring and distinguishing the information into different levels; and (2) summing up and interconnecting all of them later on again, by means of the relations already described in the ontologies themselves.

Yet, as commented above, the main contribution of this phase to the whole architecture is that it enables the model to handle the annotations from any tool, irrespective of the levels to which they pertain and the schemes (or the tagsets) employed for their generation. After the document being annotated is processed in this phase, the annotations for the same phenomenon coming from all the tools will follow the same scheme and will be, thus, comparable. A major drawback of including this phase, though, is that it requires a prior study of the output scheme and the tagsets of each of the tools assembled into the architecture. Indeed, their interpretation and mapping onto the standardised tagset obtained from OntoTag's ontologies cannot be automatically determined a priori. Consequently, an ad-hoc, tool dependent standardising wrapper must be implemented for each linguistic annotation tool assembled into an implementation of the architecture.

So, to summarise, the output of this phase is another set of documents, differing from the input ones in that they are tagged according to a standardised, tool-

independent tagset and scheme (still, one document for each tool assembled into the architecture).

2.4. Decanting

A number of the linguistic annotation tools assembled into the architecture might tag at more than just one level of linguistic description. The annotations pertaining to the same tool but to a different level have to be decanted (that is, separated according to their levels and layers or types) in a way that:

- the process of the remaining phases is not complicated; but rather
- the comparison, evaluation and mutual supplement of the results offered at the same level by different tools is simplified; and
- the different decanted results can be easily re-combined, after they have been subsequently processed.

The solution to this problem (that is, how the annotations have to be partitioned and separated) was determined empirically, after carrying out several experiments (Pareja-Lora, 2012a). Eventually, it was found that, for each annotated document coming from the tagging phase (one for each tool), two different documents have to be generated to further process morphosyntactic annotations, that is:

- one document containing both the lemmas and the grammatical category tags (L+POS);
- one consisting of the grammatical category tags and the morphological annotations (POS+M).

2.5. Merging

At this point, all the standardised and decanted annotations have to be merged in order to yield a unique, combined and multi-level (or multi-layered) annotation

for the original input document. This is the most complex part of the architecture, since it is responsible for two different tasks:

- uniting (*combining*) all the annotations that belong to the same level, but come from different tools;

- summing up and interconnecting (that is, *integrating*) the annotations that belong to different levels so as to bear a combined, integrated and unique set of annotations for the original input document.

As commented above, these two tasks are conceptually different and, thus, are considered two distinct (but intertwined) sub-phases in the architecture. Unfortunately, these two sub-phases, namely combination and integration, cannot be further described here for the sake of space.

3. The linguistic ontologies

As previously stated in Pareja-Lora (2012b, p. 326), the elements involved in linguistic annotation were formalised in a set (or network) of ontologies (OntoTag's linguistic ontologies). On the one hand, OntoTag's network of ontologies consists of:

- the Linguistic Unit Ontology (LUO), which includes a mostly hierarchical formalisation of the different types of linguistic elements (i.e., units) identifiable in a written text (across levels and layers);

- the Linguistic Attribute Ontology (LAO), which includes also a mostly hierarchical formalisation of the different types of features that characterise the linguistic units included in the LUO;

- the Linguistic Value Ontology (LVO), which includes the corresponding formalisation of the different values that the attributes in the LAO can take;

- the OIO (OntoTag's Integration Ontology), which (1) includes the knowledge required to link, combine and unite the knowledge represented in the LUO, the LAO and the LVO; and (2) can be viewed as a knowledge representation ontology that describes the most elementary vocabulary used in the area of annotation.

On the other hand, OntoTag's ontologies incorporate the knowledge included in the different standards and recommendations regarding directly or indirectly morphosyntactic, syntactic and semantic annotation so far – not discussed here for the sake of space; for further information, see Pareja-Lora (2012a, 2012b).

4. Experimentation and results

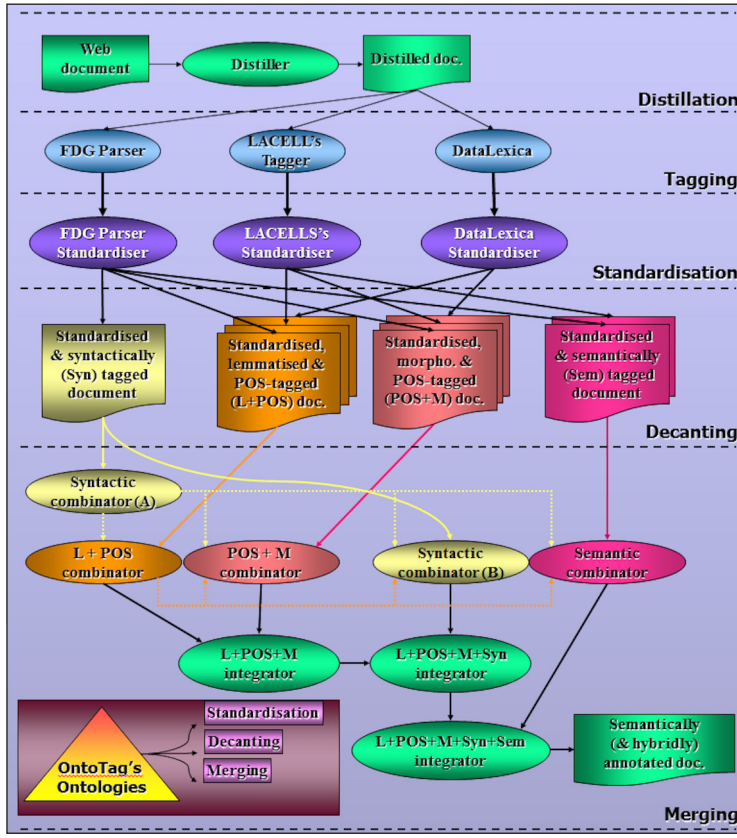
We built a small corpus of HTML web pages (10 pages, around 500 words each) from the domain of the cinema reviews. This corpus was POS tagged automatically, and its POS tags were manually checked afterwards. Thus, we had a gold standard with which we could compare the test results. Then, we used two of these ten pages to determine the rules that had to be implemented in the combination module of the prototype, following the methodology described in Pareja-Lora and Aguado de Cea (2010). Eventually, we implemented in a prototype (called *OntoTagger*) the architecture described above (see Figure 1) in order to merge the annotations of three different tools, namely Connexor's FDG Parser (henceforth FDG, http://www.connexor.com/nlplib/?q=demo/syntax), a POS tagger from the LACELL research group (henceforth LACELL, https://www.um.es/grupos/grupo-lacell/index.php), and Bitext's DataLexica (henceforth DataLexica, http://www.bitext.com/whatwedo/components/com_datalexica.html). The prototype was then tested on the remaining eight HTML pages of the corpus.

In this test, in terms of precision, the prototype (93.81%) highly outperformed DataLexica (83.82%), which actually does not provide POS tagging disambiguation; improved significantly the results of LACELL (85.68% – OntoTagger is more precise in around 8% of cases); and slightly surpassed the

results of FDG (FDG yielded a value of precision of 92.23%, which indicates that OntoTagger outperformed FDG in around 1.50% of cases).

In terms of recall, two different kinds of particular statistical indicators were devised. First, a group of indicators was calculated to show simply the difference in the average number of tokens which were assigned a more specific morphosyntactic tag by each tool being compared. For this purpose, for instance, the tags 'NC' (Noun, Common) and 'NP' (Noun, Proper) should be regarded as more specific than 'N' (Noun).

Figure 1. OntoTag's experimentation – OntoTagger's architecture



Regarding the values of the indicators in this first group, OntoTagger clearly outperformed DataLexica in 11.55% of cases, and FDG in 8.97% of cases. However, the third value of this comparative indicator shows that OntoTagger and LACELL are similarly accurate. This is due to the fact that, in fact, LACELL's morphosyntactic tags, when correct, are the most accurate of the three outputted by the three input tools. Hence, its recall can be considered the upper bound (or baseline) for this value, which is inherited somehow by OntoTagger.

On the other hand, a second group of indicators was calculated, in order to characterise the first one. Indeed, it measured the average number of tokens which are attached a more specific tag by a given tool than the others, but just in some particular cases. In these cases, the tools agreed in the assignment of the higher-level part of the morphosyntactic tag, but they did not agree in the assignment of its most specific parts. A typical example is that some tool(s) would annotate a token as 'NC', whereas (an) other one(s) would annotate it as 'NP'. Both 'NC' and 'NP' share the higher-level part of the morphosyntactic tag 'N', but not their most specific parts (respectively, 'C' = Common, and 'P' = Proper).

Regarding the values of the indicators in this second group, OntoTagger outperformed DataLexica in 27.32% of cases, and FDG in 12.34% of cases. However, once again, the third value of this comparative indicator shows that OntoTagger and LACELL are similarly accurate, which results from the same reasons described above.

Thus, to sum up, OntoTagger results were better in terms of precision than any of the annotations provided by the tools included in the experiment (only around 6% of tokens being wrongly tagged); and did not perform worse than any of them (outperforming most of them) in terms of recall.

5. Conclusions

In this paper, we have presented an annotation architecture and methodology that has helped us (1) make a set of linguistic tools (as well as their annotations)

interoperate, and (2) reduce the POS tagging error rate and/or inaccuracy of these tools. We have also presented briefly the ontologies developed to solve this interoperability problem, and shown how they were used to interlink several POS taggers together, in order to attain the goals previously mentioned. As a result, the error rate of the combined POS tagging was around 6%, whereas the error rate of the tools interlinked was in the range of 10%–15%. The resulting error rate allows including this type of technologies within language e-learning applications and environments (e.g. mobile-assisted language learning) to automatically correct the exercises and/or the errors of the learner. This should help enhance and/or improve these language e-learning scenarios, and make them more powerful and effective.

6. Acknowledgements

We would like to thank the ATLAS (UNED) research group for their constant inspiration, encouragement and support, as well as Guadalupe Aguado de Cea and Javier Arrizabalaga, without whom this research would have never been completed.

References

Borst, W. N. (1997). *Construction of engineering ontologies*. PhD thesis. Enschede. Netherlands: University of Twente.

Gruber, T. R. (1993). A translation approach to portable ontologies. *Journal on Knowledge Acquisition, 5*(2), 199-220. Retrieved from http://dx.doi.org/10.1006/knac.1993.1008

Pareja-Lora, A. (2012a). *Providing linked linguistic and semantic web annotations – The OntoTag hybrid annotation model*. Saarbrücken: LAP – LAMBERT Academic Publishing.

Pareja-Lora, A. (2012b). *OntoTag: a linguistic and ontological annotation model suitable for the semantic web*. PhD thesis. Madrid: Universidad Politécnica De Madrid. Retrieved from http://oa.upm.es/13827/

Pareja-Lora, A., & Aguado de Cea, G. (2010). Ontology-based interoperation of linguistic tools for an improved lemma annotation in Spanish. In *Proceedings of the 7th Conference on Language Resources and Evaluation (LREC 2010)* (pp. 1476-1482). Valletta, Malta: ELDA.

Urbano-Mendaña, M., Corpas-Pastor, G., & Mitkov, R. (2013). NLP-enhanced self-study learning materials for quality healthcare in Europe. In *Proceedings of the "Workshop on optimizing understanding in multilingual hospital encounters". 10th International Conference on Terminology and Artificial Intelligence (TIA'2013)* (pp. 29-32). Paris, France: Laboratoire d'Informatique de Paris Nord (LIPN).

Published by Research-publishing.net, not-for-profit association
Dublin, Ireland; Voillans, France, info@research-publishing.net

New perspectives on teaching and working with languages in the digital era
Edited by Antonio Pareja-Lora, Cristina Calle-Martínez, Pilar Rodríguez-Arancón

Typeset by Research-publishing.net
Cover design and frog picture by © Raphaël Savina (raphael@savina.net)

ISBN13: 978-1-908416-34-6 (Paperback - Print on demand, black and white)
Print on demand technology is a high-quality, innovative and ecological printing method, with which the book is never 'out of stock' or 'out of print'.

ISBN13: 978-1-908416-35-3 (Ebook, PDF, colour)
ISBN13: 978-1-908416-36-0 (Ebook, EPUB, colour)

Legal deposit, Ireland: The National Library of Ireland, The Library of Trinity College, The Library of the University of Limerick, The Library of Dublin City University, The Library of NUI Cork, The Library of NUI Maynooth, The Library of University College Dublin, The Library of NUI Galway.

Legal deposit, United Kingdom: The British Library.
British Library Cataloguing-in-Publication Data.
A cataloguing record for this book is available from the British Library.

Legal deposit, France: Bibliothèque Nationale de France - Dépôt légal: mai 2016.